UC3M AT TRECVID 2010 SEMANTIC INDEXING TASK

I. González-Díaz, V. Gómez-Verdejo, F. Díaz-de-María and J. Arenas-García

Department of Signal Theory and Communications
Universidad Carlos III de Madrid, Leganés, Spain

ABSTRACT

This paper describes the experiments carried out by the UC3M team for the TRECVID 2010 high-level feature extraction task. In our previous participations in TRECVID, we have developed a modular system to facilitate the testing of several functionalities. This year we have selected a simple system configuration and we have added some elements expected to provide an additional advantage. For instance, 1) different kinds of histogram based features have been included in the system, both with a late fusion scheme and with a spatial pyramid matching configuration; 2) according to the nature of the low level features, different kernels have been used to train a set of non-linear SVMs; and, 3) to combine the outputs of the trained SVMs, two different fusion strategies have been taken into account, an average combination and a linear 1-norm SVM. Additionally, one of our runs exploits the relationships between the different categories by taking into account the provided taxonomy relationships, thus modifying the final system output for some categories. To check the advantages provided by these new system elements, we have submitted the following runs for their evaluation in TRECVID 2010:

- **RUN 1** (*"F A UC3M 1 1"*): this run is the baseline system configuration, where the local features have been included with a Spatial Pyramid Matching scheme and an average combination of the SVMs outputs has been considered.
- **RUN 2** (*"F A UC3M 2 2"*): this run has the same configuration as RUN 1, but histogram based features have been included with a late fusion configuration.
- **RUN 3** (*"F A UC3M 3 3"*): this run replicates the configuration of RUN 2, but a 1-norm SVM has been used for the late fusion stage.
- **RUN 4** (*"F A UC3M 4 4"*): this last run modifies the final output of RUN 3 for a subset of classes according to the provided taxonomy relationships.

The four submitted runs have achieved average InfAP values from 0.0457 (RUN 1) to 0.0528 (RUN 3). Thus, our best run performed 24 out of all 87 submitted runs for this task.

1. INTRODUCTION

During our previous participations in the TRECVID high-level feature extraction (HFE) competition, we have analyzed different configuration alternatives for the components of a modular system; for instance, we have studied the advantages provided by different classification technologies, validation criteria and different fusion strategies. This year, we keep some of the elements that worked best during previous years, using a HFE system consisting of three stages: (1) a low level feature extraction layer, (2) a set of non-linear SVMs to carry out the supervised learning step and (3) a final fusion stage. To improve the performance of the system, this year we have incorporated and analyzed the performance of some new elements, namely:

- *Local features*: apart from typical Keyframe and Grid low-level features, this year we have considered a wider set of local features. Concretely, we have extracted SIFT descriptors over three different kind of local regions: circular patches in a dense grid (DSIFT), and salient affine regions using the Harris Affine Detector (HAR-SIFT) and the Hessian Detector (HES-SIFT). Furthermore, a Multi-resolution analysis with spatial grids of 1x1, 1x3 and 3x1 has been included in the system.
- *Spatial Pyramid Matching*: in addition to the basic approach (one histogram-one classifier), we have also experimented with a Spatial Pyramid approach that computes kernels at each resolution and fuses them using a kernel weighting technique.
- *Kernel selection*: to train the non-linear SVMs we have considered two kinds of kernels according to the low level features which are used as the SVMs inputs. In particular, a Histogram Intersection Kernel [1] has been used for the histogram based features and a Gaussian Kernel for the remaining low level features.

This work is part of the i3media Project (CDTI 2007 1012) and is partially funded by the Centro para el Desarrollo Tecnológico Industrial (CDTI), within the Ingenio 2010 Program.

- *Fusion scheme*: Late fusion of SVM outputs is carried out by using linear combinations following one of these two possibilities: either we use the average of all SVM outputs, or we recur to a linear 1-norm SVM. The latter scheme provides a sparse solution, so an automatic feature selection process is implicitly carried out during the learners fusion.

- *Taxonomy*: finally, we have used the given taxonomy for the original 130 categories, including a final step which changes the order of the clips classified as positive for a given category using the outputs of some other related categories.

Among the different trained runs, we have selected the four that provided better results (according to our validation tests) and, at the same time, allowed us to check the advantages or disadvantages provided by the above elements.

The remainder of this paper is organized as follows: The next section presents the Low Level Features that have been extracted from the video data. In Section 3, we describe in detail the different configurations of our system and we present the four submitted runs. Next, experimental results are analyzed in Section 4. Finally, Section 5 summarizes the conclusions.

2. LOW LEVEL FEATURE EXTRACTION

This section describes the set of low-level features that has been included in our concept detection system. In general, the low-level descriptors have been organized in three levels or granularities, namely (a) *Keyframe Level features*, that describe image content on each keyframe, (b) *Regular Grid features*, which apply some kind of spatial regionalization by dividing the image into a regular grid, and (c) *Local features* that detect and describe specially discriminant areas in images. Next, we provide a brief description of each of them.

2.1. Keyframe level features

Each keyframe extracted from the video content is described by means of several still image descriptors. To reduce the number of learning machines in our system, we have designed an early fusion stage that combines features following a simple categorization: *color features* and *texture features*. This fusion stage simply concatenates features using equal weights and, then, normalizes each element on the resulting vector to have zero mean and unit variance. The Color vector include several MPEG-7 color descriptors (Color Structure, Scalable Color and Color Layout) and Color Correlograms and AutoCorrelograms, whereas the Texture vector is composed of other MPEG-7 descriptors (Homogeneous Texture, Texture Browsing), as well as Gray-Level Co-Ocurrence Matrices, Gabor Wavelets and Edge Histograms.

2.2. Regular Grid features

Each keyframe has been divided using a regular grid of type 4x3. Then, each cell has been annotated using two compact descriptors: *Color Moments* (CM) (up to 3rd-order) in HSV color space, and *Gabor Wavelets* (GW) with two scales and four orientations.

2.3. Local Features. The bag-of-words model

Bag-of-Words (BoW) models have shown exceptional performance and constitute the most prevalent approach for concept detection in audiovisual content. These models were initially proposed for text retrieval and later used in computer vision, where the traditional “document” became an image and the “words” were associated with visual words that describe the content of local patches. BoW models make a simplifying assumption on the data distribution in a image, which is simply considered as an unordered collection of visual words that describe the appearance of local regions. Good examples of BoW models can be found in both discriminative [2] [3] and generative frameworks [4] [5]. Originally, these models did not take into account the spatial location of the visual words in an image, what, obviously, limits their performance. More recently, some spatial constraints have been proposed for BoW models to benefit from spatial discrimination to some extent. In particular, the discriminative approach called Spatial Pyramid Matching ([6],[7],[8]) attains improved classification performance by computing image histograms at different spatial levels and a weighted kernel over each level.

We have experimented with several parameters of the traditional bag-of-words model. Next we provide a brief description of every module in our bag-of-words approach:

- ***Detection of local regions***: Two affine covariant region detectors have been used to detect salient regions in each image: Hessian Affine Detector and Harris Detector. The interested reader is referred to [9] for a complete description of these detectors. Moreover, a dense grid with two scales has also been used, which yielded overlapped circular patches with radius 8, and 16, organized in a regular 6 pixel spaced grid.

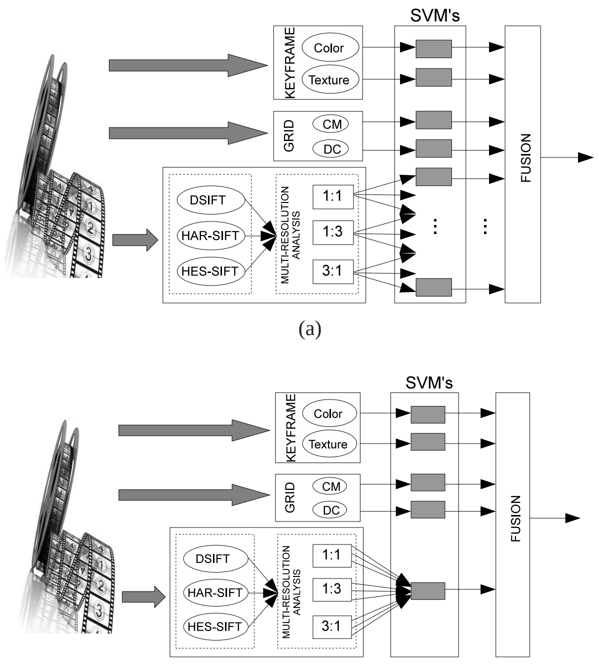
- ***Local feature extraction***: Local features are extracted from each local patch in every image. The texture appearance of each region is described by means of a 128-dimensional SIFT descriptor [10]. It is noteworthy that each detector (Harris, Hessian and Dense Grid) provides its particular set of descriptors per image that are treated separately by the bag-of-words model.

- ***Visual Vocabulary***: Once the local features have been extracted, a bag-of-words model is computed for each detector. The k-means clustering algorithm has been used to compute the M codewords that best represent the local features of the reference image set. In our case, a vocabulary size of $M = 4000$ has been used for each detector. Following the aforementioned approach three visual vocabularies have been computed to represent words coming from each of the detectors.

- ***Histograms of visual words***: Every image is then vector-quantized so that each region descriptor is assigned to its closest codeword and a normalized histogram of words is computed. In particular, we have employed a soft-assignment technique that computes a gaussian kernel between a word and each of the codewords and increments the positions of the histogram with the obtained values.

- ***Multi-resolution analysis***: Finally, each image is explored at several spatial granularities so that histograms at different spatial levels are generated. In particular, we have computed histograms using the following grids: 1x1, 1x3 and 3x1. Furthermore, a spatial pyramid kernel has been also utilized to perform a multi-resolution analysis of the words distribution along the image.

Hence, the combination of three detectors and three spatial grids gives place to nine low-level descriptors to which we should add the spatial pyramid kernel as potential inputs for the classification stage.



(a)

(b)

Fig. 1. General system architecture when local features are employed in a late fusion scheme (Subfigure (a)) and in Spatial Pyramid Matching scheme (Subfigure (b)).

3. HIGH LEVEL FEATURE EXTRACTION

In this section we are going to describe how the different runs have been created. As we have already explained, we have considered, as our starting point, a modular system architecture made up of three processing steps: (1) a low level feature extraction layer, (2) a set of supervised learning machines, and (3) a final fusion stage. This year we have done without a feature selection stage, but as we will show later, one of the selected fusion schemes is able to automatically carry out the feature selection process. Finally, we will present the four runs submitted to TRECVID, indicating which elements differentiate one run from the others.

3.1. Early/late fusion of low-level features

Once all the low level features have been extracted from the video data (as described in Section 2), they are employed to train an SVM which solves, for each category, the desired classification problem. Keyframe and Grid features have been directly used as inputs to the classifiers; however, the histograms of local features have been grouped according to two different architectures (see Figure 1):

- *Late fusion*: each feature becomes an input of a different SVM.

- *Early fusion*: all local features described in Section 2.3 have been grouped to become the input of a unique SVM. This scheme corresponds to the well-known Spatial Pyramid Matching ([6, 7, 8]) which attains improved classification performance by computing image histograms at different spatial levels and a weighted kernel over each level.

3.2. SVMs classifiers

After the low level feature extraction, the next stage of our system consists of a set of non-linear SVMs trained with the LIBSVM toolbox [11]. These SVMs have been trained with a maximum of 8000 data, using as many positive data as available, and allowing a maximum of 10 negative instances per each positive one. The free SVM parameters have been adjusted with a cross-validation procedure using the Average Precision (AP) as the validation criterion.

When Keyframe or Grid low level features are considered as SVM inputs, the Gaussian kernel is used,

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}\right)$$

σ being the kernel width to be adjusted during the SVM cross-validation process.

However, when dealing with local features, it can be more useful to consider other kernels. In particular, due to the fact that local descriptors are histogram based features, it seems reasonable to use histogram-based kernels, such as the Histogram Intersection Kernel [1]:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{d} \sum_{k=1}^{d} \min \left\{ x_i^{(k)}, x_j^{(k)} \right\}$$

where d is the total number of bins of histograms $\mathbf{x}_i$ and $\mathbf{x}_j$ (i.e., d is the dimension of local descriptors). This kernel has the additional advantage of being computationally faster than the Gaussian one. Furthermore, it has no parameters to be adjusted, what implies also important computational savings during the cross-validation. The overall CPU-time saving due to the previous kernel properties are specially convenient when dealing with high dimensional features, such as histograms of local descriptors.

3.3. Fusion stage

Finally, the SVM outputs have been linearly combined in order to obtain the global system output. To carry out this combination, we have considered two schemes:

- An average combination of all SVM outputs: in this case the final system output, $f(\mathbf{x})$, is obtained by directly averaging SVMs outputs, i.e.,

  $$f(\mathbf{x}) = \frac{1}{T} \sum_{t=1}^{T} o_t(\mathbf{x})$$

  where $o_t(\mathbf{x})$ is the output of the t-th SVM for the input low level feature **x**. Note that according to previous results reported in the TRECVID HFE task, and also according to our own experiments, more sophisticated trainable linear combinations do not provide additional gains with respect to the very simple average of all outputs.

- A sparse linear SVM: this combination scheme uses a linear 1-norm SVM [12] to obtain a set of weights to combine the SVMs outputs. In this way, the final system output is given by:

  $$f(\mathbf{x}) = \mathbf{w}^T \mathbf{o}(\mathbf{x}) + b$$

  where $\mathbf{w} = [w_1, \ldots, w_T]^T$ is the combination weight vector, $\mathbf{o}(\mathbf{x}) = [o_1(\mathbf{x}), \ldots, o_T(\mathbf{x})]^T$ is a vector containing the SVMs outputs and b is the bias term. To learn these parameters, in such a way that a maximum margin solution is provided, the 1-nom SVM solves the following optimization problem:

$$\begin{aligned} \text{min} \quad & \|\mathbf{w}\|_1 + \frac{C}{N} \sum_{i=1}^{N} \xi_i \\ \text{s.t.} \quad & y_i \left(\mathbf{w}^T \mathbf{o} \left(\mathbf{x}_i \right) + b \right) \geq 1 - \xi_i; \qquad i = 1, \ldots, N \\ & \xi_i \geq 0; \qquad i = 1, \ldots, N \end{aligned} \tag{1}$$

where slack variables ξ_i are introduced to allow the SVM outputs for some training data to be misclassified or to lie inside the classifier margin, C is a constant that controls the tradeoff between the structural and empirical risk terms, and y_i is the label (desired output) associated to training pattern $\mathbf{x}_i$.

The main difference between this formulation and a classical 2-norm SVM, relies on the fact that an 1-norm regularization term has to be minimized. This regularizer presents singularity points whenever any of the components of **w** is zero, what tends to nullify some of the solution weights, thus favoring sparse solutions. Therefore, this fusion scheme implicitly provides a feature selection criterion, since all SVMs in the first stage whose associated weight is set to zero at the combination stage by the 1-norm SVM can be removed from the system.

3.4. Sorting out positive clips using a predefined taxonomy

One of the main novelties of this year's task is the availability of a list with predefined relationships between classes, in the form "class A implies class B". In principle, this can be used to refine the results for class B, since clips which are labeled as positive for class A should also be positive for class B. However, some initial validation results suggested this was not a good strategy, since typically the classifiers for the broader classes worked better than those for more specific concepts, probably due to the availability of a larger number of positive instances.

Therefore, we have tried quite a different approach, consisting in using the output of the classifier for class B, to sort out the first 1000 positive instances already detected for class A. In this way, we expect that the results for class A can benefit from the wider dataset used for learning concept B. We have also established a restriction to apply the previous procedure, based on a maximum number of positive instances in class A (i.e., we only apply the output refining procedure when the performance of classifier A is expected to be very poor). For instance, category "Ground Vehicles" (which has 2236 positive instances) has been used to modify the output of "Bus" (with only 32 positive data).

Following this criterion, we modified the outputs for 24 out of the 130 categories and, among them, for 3 out of the 30

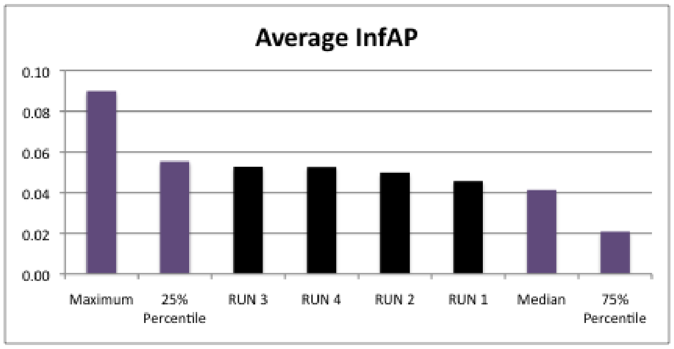


Fig. 2. Average InfAP of all our submitted runs. Results are shown in comparison with the best performing run, median, and 25% and 75% percentiles.

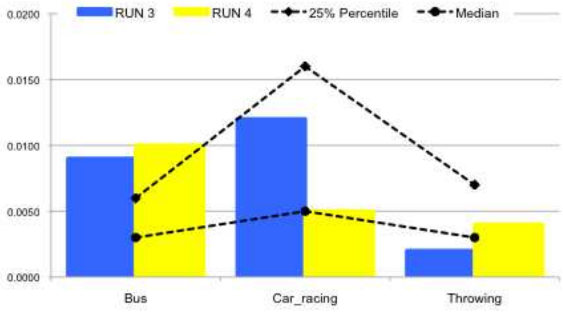


Fig. 3. InfAP of RUNS 3 and 4 over the categories where the taxonomy has been applied. TRECVIDs 25th percentile and median are also included for comparison.

categories that were finally evaluated.

3.5. Runs submitted to TRECVID 2010

The combination of the above elements of the system have provided us a set of possible runs to be submitted to TRECVID 2010, analyzing their final validation AP values, the following configurations have been selected for evaluation at TRECVID:

- **RUN 1** (*"F A UC3M 1 1"*): the first run is the baseline system configuration, where the local features have been included with a Spatial Pyramid Matching scheme and an average combination of the SVMs outputs has been considered. No use of taxonomies has been incorporated into this run.

- **RUN 2** (*"F A UC3M 2 2"*): this run has the same configuration as RUN 1, but histogram based features have been included with a late fusion configuration.

- **RUN 3** (*"F A UC3M 3 3"*): this run replicates the configuration of RUN 2, replacing the average late fusion strategy by an 1-norm SVM.

- **RUN 4** (*"F A UC3M 4 4"*): this last run modifies the outputs of RUN 3 for 24 classes using the available taxonomy.

4. PERFORMANCE EVALUATION

In this section we evaluate the performance of all four runs in terms of InfAP. To start with, Figure 2 illustrates the achieved InfAP averaged over the 30 high-level concepts that have been selected (among the original 130) for its evaluation. The best result (maximum), median and the 25% and 75% percentiles are also shown in the figure as a reference for comparison. Our submitted runs have achieved average InfAP values from 0.0457 (RUN 1) to 0.0528 (RUN 3), what places all our designs in the second quartile of all TRECVID 2010 submitted runs. To be more specific, our best performing run was ranked 24 among all 87 submitted runs.

The achieved infAP results confirmed the derived hypotheses during the design and validation phases. The baseline was clearly outperformed by all other designs, and in fact the 1-norm SVM, with implicit feature selection, showed to be much more effective than simple averaging for fusing the ouptuts of the SVM networks. Using Spatial Pyramid Matching early fusion for the local descriptors provided unsatisfactory results, with RUN 2 already providing significantly better performance than the baseline.

As it was expected, results from RUNS 3 and 4 did not differ significantly, since RUN 4 only provided different outputs for 3 out of the 30 validated concepts. Figure 3 displays infAP for classes "bus", "car racing" and "throwing". We can see that our procedure for exploiting taxonomy gave some improvements in two of the classes, while significantly degrading the recognition accuracy for the "car racing" class. Overall, RUN 4 performed slightly worse than RUN 3 on average, but it would be interesting to study results for other classes where the taxonomy was also exploited.

Figure 4 provides detailed information about the achieved infAPs over the 30 evaluated concepts. We can check that our previous discussion about the different runs is essentially still valid when analysing each of the class separately. However, in this case we can also observe that for many of the concepts RUN 2 actually outperformed RUN 3, which shows that a simple average late fusion process is good enough in many cases. Apart from this, we can also see that our designs performance is close to the 25% percentile not just on average, but also when analyzing each of the high level concepts individually.

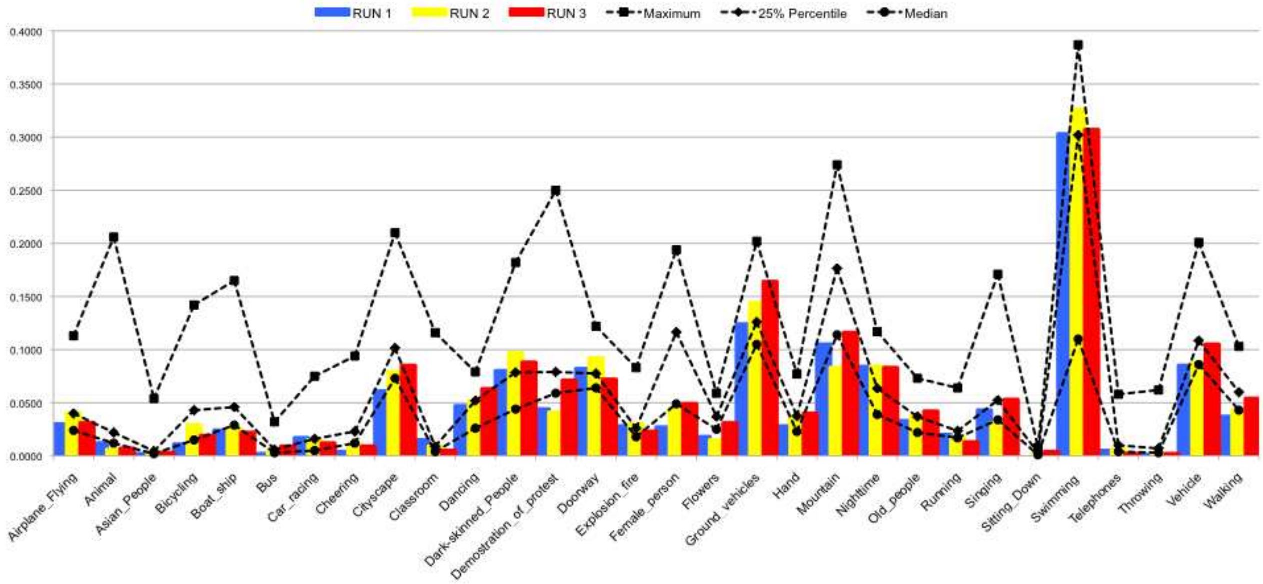


Fig. 4. InfAP per class of RUNS 1, 2 and 3. TRECVIDs maximum, 25th percentile and median are also included as a reference for comparison.

5. CONCLUSIONS

In this paper, we have presented the research work of the UC3M team at the TRECVID 2010 semantic indexing task. This year, the most salient novel elements in our classification system were:

- New Local Descriptors, including also a Spatial Pyramid Matching early fusion strategy
- Mixed used of the Gaussian and Histogram Intersection Kernel (for histogram-like features)
- 1-norm linear SVM for late fusion with implicit feature selection
- Positive items re-sorting based on the provided taxonomy

The submitted runs could assess the advantages of the three first items, while the impact of taxonomy exploitation is unclear, due to the small number of classes where this mechanism was implemented that were selected for the task evaluation.

Our submitted runs have achieved average InfAP values from 0.0457 (RUN 1) to 0.0528 (RUN 3), what places all our designs in the second quartile of all TRECVID 2010 submitted runs, and out best performing system 24 among all 87 TRECVID submitted runs.

6. REFERENCES

[1] A. Barla, F. Odone, and A. Verri, "Histogram intersection kernel for image classification," sep. 2003, vol. 3, pp. 513–516.

[2] C. Wallraven, B. Caputo, and A. Graf, "Recognition with local features: the kernel recipe," in *IEEE International Conference on Computer Vision*, Oct. 2003, pp. 257–264 vol.1.

[3] K. Grauman and T. Darrell, "The pyramid match kernel: discriminative classification with sets of image features," in *IEEE International Conference on Computer Vision*, Oct. 2005, vol. 2, pp. 1458–1465 Vol. 2.

[4] Thomas Hofmann, "Unsupervised learning by Probabilistic Latent Semantic Analysis," *Machine Learning*, vol. 42, no. 1/2, pp. 177–196, 2001.

[5] David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty, "Latent Dirichlet Allocation," *Journal of Machine Learning Research*, vol. 3, pp. 2003, 2003.

[6] C. Schmid S. Lazebnik and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006, vol. 2, pp. 2169–2178.

[7] Anna Bosch, Andrew Zisserman, and Xavier Munoz, "Representing shape with a spatial pyramid kernel," in *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, New York, NY, USA, 2007, pp. 401–408, ACM.

[8] M. Varma and D. Ray, "Learning the discriminative powerinvariance trade-off," in *In ICCV*, 2007.

[9] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," *International Journal of Computer Vision*, vol. 65, no. 1/2, pp. 43–72, 2005.

[10] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[11] C. Chang and C. Lin, "LIBSVM: a library for Support Vector Machines," 2001, Software available at http://www.csie.ntu.edu.tw/c̃jlin/libsvm.

[12] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," in *Advances in Neural Information Processing Systems 16*, Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, Eds., pp. 49–56. MIT Press, Cambridge, MA, 2004.